In the claims

All of the claims standing for examination are reproduced below. Claims 1, 7, 13 and 19 are amended in this response.

1. (Currently amended) In a distributed processor system wherein a first and a second protocol operating on individual ones of a first plurality of processors operating within a single data packet router are involved in independently generating or amending data for a single database, and wherein each of the first plurality of processors maintains a copy of the database, a method for synchronized maintenance and distribution of the database, comprising the steps of:

(a) registering each of the first plurality of processors with at least one other of the first plurality of processors, creating client-server pairs operating within the single data packet router, in an arrangement that each of the plurality of processors either runs or is registered with a processor running both the first and second protocols; and

(b) sharing the generated or amended data from the servers to the registered clients, such that each of the first plurality of processors receives generated or amended data from both the first and second protocols.

2. (Original) The method of claim 1 wherein the system comprises a second plurality of processors upon which the first and second protocol do not run, and further comprising a step (c) for registering each of the second plurality of processors with at least one of the first plurality of processors, creating client-server pairs between individual ones of the first and second plurality of processors, and a step (d) for sharing at least a subset of the database from the

servers in the first plurality of processors to the clients in the second plurality of processors.

3. (Original) The method of claim 2 comprising a third plurality of processors upon which the protocols do not run, and further comprising a step (e) for registering each of the third plurality of processors with individual ones of the second plurality of processors, creating client-server pairs between individual ones of the second and third plurality of processors, enabling clients in the third plurality of processors to receive copies of the subset of the database.

4. (Original) The method of claim 3 wherein, in one or more of steps (a), (c) and (e) clients register with a second processor to create a redundant server-client relationship for fault tolerance.

5. (Original) The method of claim 4 wherein a client treats the two servers with which it registers as a primary and a secondary server, and communicates only with the primary server as long as the primary server remains capable, and further comprising a step for activating the secondary server in the event the primary server fails.

6. (Original) The method of claim 5 wherein, upon activation of the second server, a copy of the database is sent to the client, which compares that copy with its own copy, determines the difference, and uses only the difference in further propagation of copies.

7. (Currently amended) A distributed processor system comprising:

a first plurality of processors operating within a single data packet router, each maintaining a copy of a single database; and

a first and a second protocol operating on individual ones of the first plurality of processors, the protocols independently generating or amending data for the single database;

characterized in that each of the first plurality of processors registers with at least one other of the first plurality of processors, creating client-server pairs operating within the single data packet router, in an arrangement that each of the plurality of processors either runs or is registered with a processor running both the first and second protocols, and the servers of the client-server pairs share the generated or amended data with the clients, such that each of the first plurality of processors receives generated or amended data from both the first and second protocols.

8. (Original) The system of claim 7 comprising a second plurality of processors upon which the first and second protocol do not run, wherein each of the second plurality of processors registers with at least one of the first plurality of processors, creating client-server pairs between individual ones of the first and second plurality of processors, and at least a subset of the database is shared from the servers in the first plurality of processors to the clients in the second plurality of processors.

9. (Original) The system of claim 8 comprising a third plurality of processors upon which the protocols do not run, wherein each of the third plurality of processors register with individual ones of the second plurality of processors, creating client-server pairs between individual ones of the second and third plurality of processors, enabling clients in the third plurality of processors to receive copies of the subset of the database.

10. (Original) The system of claim 9 wherein clients register with a second processor to create a redundant server-client relationship for fault tolerance.

11. (Original) The system of claim 10 wherein a client treats the two servers with which it registers as a primary and a secondary server, communicates only with the primary server as long as the primary server remains capable, and activates the secondary server in the event the primary server fails.

12. (Original) The system of claim 11 wherein, upon activation of the second server, a copy of the database is sent to the client, which compares that copy with its own copy, determines the difference, and uses only the difference in further propagation of copies.

13. (Currently amended) In a single data packet router wherein a first and a second routing protocol generating routing data operate on individual ones of a first plurality of processors, and wherein each of the first plurality of processors maintains a copy of the routing table, a method for synchronized maintenance and distribution of the routing table and a forwarding table subset of the routing table, comprising the steps of:

(a) registering each of the first plurality of processors with at least one other of the first plurality of processors, creating client-server pairs operating within the single data packet router, in an arrangement that each of the plurality of processors either runs or is registered with a processor running both the first and second routing protocols; and

(b) sharing the routing data from the servers to the registered clients, such that each of the first plurality of processors receives the routing data from both the first and second routing protocols.

14. (Original) The method of claim 13 wherein the data packet router comprises a second plurality of processors upon which the first and second protocol do not run, and further comprising a step (c) for registering each of the second plurality of processors with at least one of the first plurality of processors, creating client-server pairs between individual ones of the first and second plurality of processors, and a step (d) for sharing a forwarding table subset of the routing table from the servers in the first plurality of processors to the clients in the second plurality of processors.

15. (Original) The method of claim 14 comprising a third plurality of processors upon which the protocols do not run, and further comprising a step (e) for registering each of the third plurality of processors with individual ones of the second plurality of processors, creating client-server pairs between individual ones of the second and third plurality of processors, enabling clients in the third plurality of processors to receive copies of the forwarding table.

16. (Original) The method of claim 15 wherein, in one or more of steps (a), (c) and (e) clients register with a second processor to create a redundant server-client relationship for fault tolerance.

17. (Original) The method of claim 16 wherein a client treats the two servers with which it registers as a primary and a secondary server, and communicates only with the primary server as long as the primary server remains capable, and further comprising a step for activating the secondary server in the event the primary server fails.

18. (Original) The method of claim 17 wherein, upon activation of the second server, a copy of the routing table of forwarding table is sent to the client, which

compares that copy with its own copy, determines the difference, and uses only the difference in further propagation of copies.

19. (Currently amended) A data packet router comprising:

a first plurality of processors operating within a single data packet router, each maintaining a copy of a routing table; and

a first and a second protocol operating on individual ones of the first plurality of processors, the protocols independently generating or amending routing data for the routing table;

characterized in that each of the first plurality of processors registers with at least one other of the first plurality of processors, creating client-server pairs operating within a single data packet router, in an arrangement that each of the plurality of processors either runs or is registered with a processor running both the first and second routing protocols, and the servers of the client-server pairs share the generated or amended routing data with the clients, such that each of the first plurality of processors receives generated or amended routing data from both the first and second routing protocols.

20. (Original) The router of claim 19 comprising a second plurality of processors upon which the first and second routing protocols do not run, wherein each of the second plurality of processors registers with at least one of the first plurality of processors, creating client-server pairs between individual ones of the first and second plurality of processors, and at least a forwarding table subset of the routing table is shared from the servers in the first plurality of processors to the clients in the second plurality of processors.

21. (Original) The router of claim 20 comprising a third plurality of processors upon which the routing protocols do not run, wherein each of the third plurality of

processors register with individual ones of the second plurality of processors, creating client-server pairs between individual ones of the second and third plurality of processors, enabling clients in the third plurality of processors to receive copies of the forwarding subset of the routing database.

22. (Original) The router of claim 20 wherein clients register with a second processor to create a redundant server-client relationship for fault tolerance.

23. (Original) The router of claim 22 wherein a client treats the two servers with which it registers as a primary and a secondary server, communicates only with the primary server as long as the primary server remains capable, and activates the secondary server in the event the primary server fails.

24. (Original) The router of claim 23 wherein, upon activation of the second server, a copy of the routing or forwarding table is sent to the client, which compares that copy with its own copy, determines the difference, and uses only the difference in further propagation of copies.